

## Research Areas

---

Programming language design and implementation, compiler construction, software testing and debugging automation, assertion languages, visual programming, component-based software engineering, computer security.

## Research Description

---

**Programming language implementation tools:** We have designed a compiler writing language RIGAL. This is a powerful and convenient tool for rapid prototyping of programming language processors. Examples of implemented language processors include PASCAL subset assertion checker, AWK assertion checker, Miranda abstract machine, FrameMaker to BBN/Slate filter, monitor generator for the Unicon language, and, of course, the RIGAL compiler itself.

**Testing and debugging automation tools:** The project includes assertion language development and prototype design for debugging automation. The approach is based on the notion of event grammar that is a general basis for program behavior model. A language based on event patterns and computations over target program execution history can be used for assertion checking, performance measurement, program profiling, debugging queries for both sequential and parallel programs, and program behavior visualization.

**The design of the V visual data flow language:** We suggest a solution for iterative processing in data flow diagrams based on the notion of a conditional data flow switch, and a specialized iterative construct based on pattern matching for vectors, matrices, and multisets. Both of these constructs can be seamlessly incorporated into a data flow visual programming language. We demonstrate how these constructs may be used to reveal the spatial/temporal dualism of data streams.

**Component-based software design:** This is a collaborative project supported by US ONR. The objectives are to develop a uniform meta-model for component-based distributed software design, Quality of Service metrics, and generative domain models.

**Computer security:** In collaboration with colleagues from NPS, we are working on the methods and tools for intrusion detection and countermeasures, based on automatic system kernel instrumentation.

## Relevance to DoN/DoD

---

Software design automation tools and expressive programming languages are necessary to improve quality and to facilitate design of complex software systems for DoN/DoD needs.

## Recent Publications

---

M.Auguston, "RIGAL - a programming language for compiler writing", Lecture Notes in Computer Science, Springer Verlag, vol.502, 1991, pp.529-564.

- M. Auguston, "Building Program Behavior Models,"  
ECAI-98 Workshop on Spatial and Temporal Reasoning, Brighton, England, August 23-28, 1998, pp.19-26.
- M. Auguston, Tools for Program Dynamic Analysis, Testing, and Debugging Based on Event Grammars, in  
Proceedings of the 12th International Conference on Software Engineering and Knowledge Engineering, Chicago,  
USA, July 6-8, 2000, pp.159-166
- M. Auguston, C. Jeffery and S. Underwood, "A Framework for Automatic Debugging," Proceedings of the IEEE  
17th International Conference on Automated Software Engineering, ASE'02, Edinburgh Scotland, September 2002,  
pp. 217-222
- M. Auguston and A. Delgado, "Iterative Constructs in the Visual Data Flow language", Proceedings of the IEEE  
Symposium on Visual Languages VL97, Capri, Italy, September 1997, IEEE Computer Society, pp.152-159.
- M. Auguston, V. Berzins, and B. Bryant, "Visual Meta-Programming Language," Proceedings of OOPSLA 2001  
Workshop on Domain-Specific Visual Languages, October 14, 2001, pp.69-82, Tampa, Florida
- J. Bret Michael, M. Auguston, N. Rowe, R. Riehle, "Software Decoys: Intrusion Detection and Countermeasures",  
Proc. IEEE Workshop on Information Assurance, United States Military Academy, West Point, NY, June 2002, pp.  
130-138.